High-**R**ise **D**atabase-**A**ssisted **D**esign **1.1** (**HR_DAD_1.1**) Software

User's Manual

Developed by Seymour M. J. Spence

Building and Fire Research Laboratory
National Institute of Standards and Technology

Current version of software available at www.nist.gov/wind

Disclaimer: Certain trade names or company products are specified in this document to specify adequately the procedure used. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the product is the best available for the purpose. The "stand-alone" version of this software requires installation of the MATLAB[1] Component Runtime (MCR) Libraries provided by The MathWorks, Inc. The author's limited rights to the deployment of this program are limited by a license agreement between NIST and The MathWorks. The license agreement can be found at www.mathworks.com/license/. The author, NIST, and The MathWorks and its licensors are excluded from all liability for damages or any obligation to provide remedial actions.

1. Introduction

This User's Manual is designed to assist the user of the software HR_DAD_1.1. The software calculates the response of tall buildings subjected to wind loads, including internal forces in members, member interaction formulas based on demand-to-capacity ratios, inter-story drift, and accelerations, for any specified mean recurrence interval of the wind effect being considered.

2. How to download and install the software

The **HR_DAD_1.1** software has been developed using the MATLAB® language and can be accessed via the internet site http://www.nist.gov/wind. Within the site, click the link "Wind Effects on Flexible Buildings". This opens the main page "HR_DAD_1.1 - DAD Software for High-Rise Buildings" (direct access: http://www.itl.nist.gov/div898/winds/hr_dad_1.1/hr_dad_1.1.htm). The files available for download are all in the bulleted list under the heading "*Files Available for Download*". In the following, reference is made to each set of files to be downloaded by the name of the associated bullet. From the list, first consider the "Files for **HR_DAD_1.1** software" bullet. Next to the title of the bullet item, there is a link to the self-extracting file zip file "HR_DAD_1.1.exe", which contains the stand-alone executable. To run HR_DAD_1.1 the MCRInstaller must first be installed by executing the application "MRCInstaller.exe". The HR_DAD_1.1 software can then be launched by double-clicking the file "HR_DAD.exe" within the "stand-alone" folder. This action opens the ten figure windows ('Page_Open', 'Page_Main', 'Page_One', 'Page_Two' … 'Page_Eight') that form the graphical user interface (GUI).

3. Basics of using the **HR_DAD_1.1** software

The ten figure windows (i.e., the "pages") opened above are used primarily to (1) assign values to the variables used by the **HR_DAD_1.1** software ('Page_One' through 'Page_Five'), (2) to perform the calculations ('Page_Six'). Variable values can be assigned in any order in 'Page_One' through 'Page_Five'. The variable names within **HR_DAD_1.1** are typically shown in parentheses before the input box on each page. In several instances, a saved MATLAB mat file is opened within a page to load variables that contain vectors, matrices, or 3D arrays. Help icons, , are located next to the input boxes for key variables. For a given variable, clicking on the associated help icon will open a separate window that provides information such as the variable name, the required variable size, a description of the variable and the specific organization of the variable's contents (for vector, matrix or 3D array variables).

4. How to use the Manual

This manual is organized with the aim of guiding the user through the necessary steps in order to run **HR_DAD_1.1**. This is done by first defining a simple 3D 2-story building. The input necessary for running this example in **HR_DAD_1.1** is then schematically outlined, allowing the user to get an idea of the software's operation. Each page of the graphical user interface (GUI) is then described in detail, together with what the input would be in the case of the simple 3D 2-story building.

5. Definition of the 3D 2-story building

Consider the structure in Fig. 1 in which the floors are considered rigid and no rotation is allowed between the floor slabs and the 8 columns (members 1 to 8).

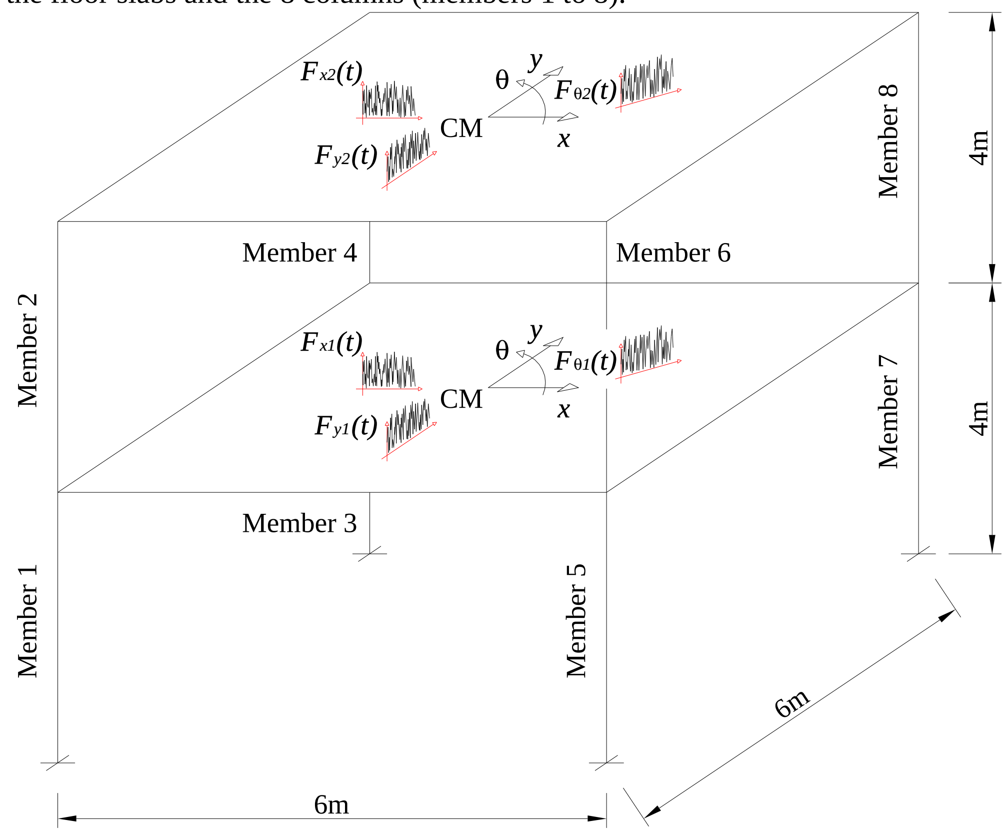


Figure 1.Simple 2 story 3D building.

6. Input for Running HR_DAD_1.1

To run **HR_DAD_1.1** information is necessary concerning the structure, the wind climate at an open site at a reasonable distance from the building, and the relation between the directional speeds at 10 m elevation at that site and the corresponding directional wind speeds (including veering effects, see Simiu and Miyata, 20006, p. 15), as follows:

1. Information related to the dynamic properties of the structure:

   § **Mass** and **mass moment of inertia** with respect to the center of mass, for each floor..

   § **Mode shapes, frequencies, and damping ratios**. The mode shapes must be referenced to the centers of mass of each floor.

2. Information for member demand-to-capacity ratio calculations:

   § **Influence coefficients,** defined as *the internal force that occurs in a given section of a given member due to a unit force applied to the center of mass of a given floor*.

3. Information relating to the structural loading. In particular **HR_DAD_1.1** is capable of considering:

- **Time varying loads** applied at the center of mass of each floor, obtained from wind tunnel tests carried out on rigid models
- **Static loads** applied to each member.

4. Information relating to the wind climate at or near the building site

- In general, large separate **databases of directional wind speeds** are needed for each type of wind occurring at the site (e.g., **hurricanes, thunderstorms, synoptic winds**) – see Grigoriu, 2008). The calculation of mean recurrence intervals of wind effects is analogous to the simple calculations discussed for wind speeds by Simiu and Miyata (2006, p. 28). An algorithm for incorporating this calculation is currently being developed.

- Information regarding the ratio between wind speeds at 10 m above ground in open terrain and their mean hourly wind speed counterparts (as affected by veering, see Simiu and Miyata, 2006, p. 15) at the top of the building. An algorithm for incorporating the effect of veering is currently being developed.

7. Output from Running HR_DAD_1.1

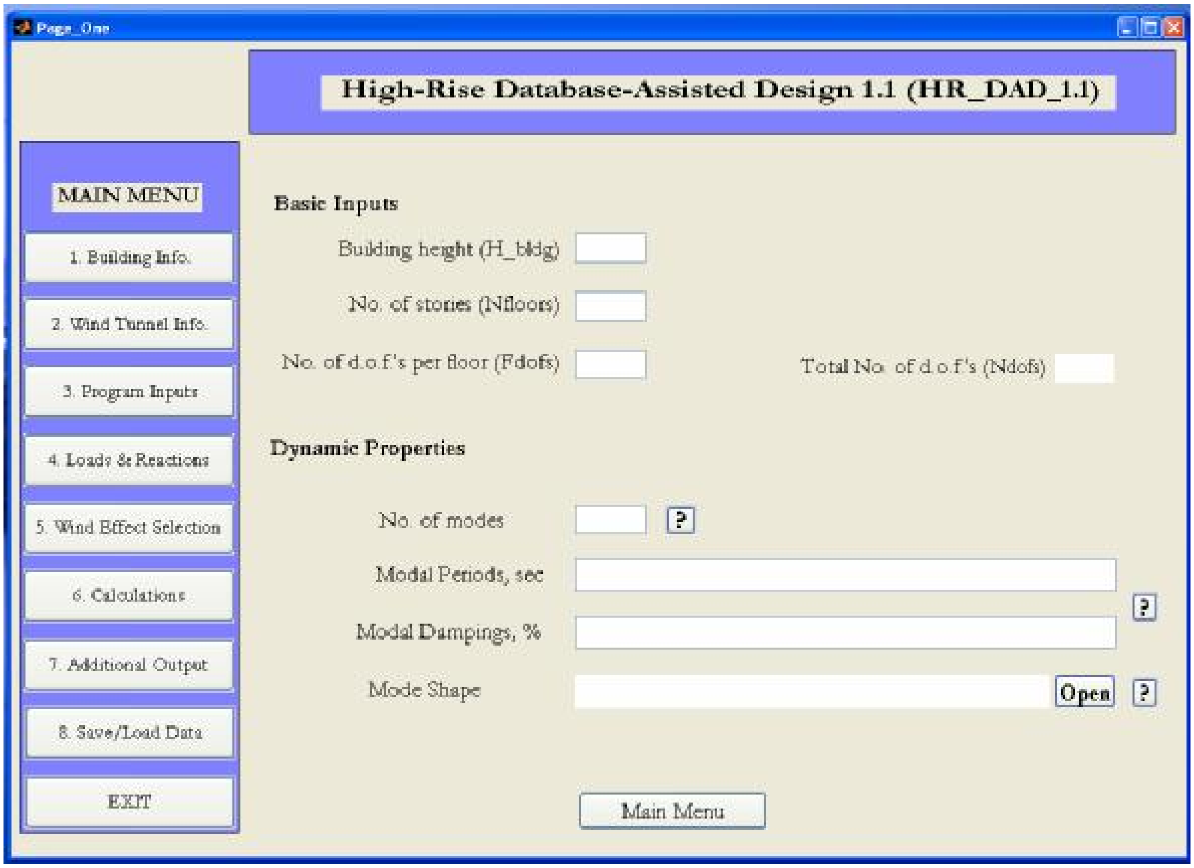
- **Peak demand/capacity ratios** for all members calculated for any number of Mean Recurrence Intervals (**MRIs**).
- **Peak inter-story drift** calculated for any number of column lines and **MRIs.**
- **Peak top floor acceleration** calculated for any number of points and any number of **MRIs.**

---

8. Details on Use of the Software

The following pages will illustrate in detail how to use the eight pages that make up the GUI of **HR_DAD_1.1**. During the general description of the necessary input, and the formatting of such input, the variables necessary for running the example building shown in figure 1 will be input under the heading 3D 2-story example.

PAGE_ONE



INPUT:

H_bldg = Height of the building in meters

3D 2-story example: total building height is 8m.

Nfloors = Number of floors composing the building

3D 2-story example: 2 floors.

Fdofs = Degrees of freedom per floor

3D 2-story example: 3.

No. of modes = Choose the number of vibrational modes to be considered in the analysis

3D 2-story example: 3 modes are chosen.

Modal Periods = assign the modal periods in seconds to each vibrational mode

3D 2-story example: if 3 modes are to be considered in the analysis and the periods are, for example, 1.7 s, 1.5 s and 0.8 s, then the following vector would be entered:

[0.6122 0.6102 0.4565]

Note: The variable must be input using square brackets as shown.

Modal damping, % = assign the modal damping as a percentage of critical.

3D 2-story example: if 3 modes are to be considered in the analysis and the modal damping is to 1.5% of critical for each mode, then the following vector should be entered;

[1.5 1.5 1.5]

Note: The variable must be input using square brackets as shown.

Mode shapes = Input the Matlab mat file containing the mode shapes.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the mode shapes must be named ***evectors***. The file containing the variable maybe named anyway desired. The variable ***evectors*** is a matrix whose columns contain the mode shapes referred to the center of mass of each floor. Each column has the x-coordinates first then the y-coordinates and finally the θ-coordinates of the mode shape.

3D 2-story example: consider the following mode shapes:

$$\text{mode shape }1=\begin{bmatrix}0.62\\1\\0\\0\\0\\0\end{bmatrix},\ \text{mode shape }2=\begin{bmatrix}0\\0\\0.62\\1\\0\\0\end{bmatrix}\ \text{and mode shape }3=\begin{bmatrix}0\\0\\0\\0\\0.62\\1\end{bmatrix}.$$

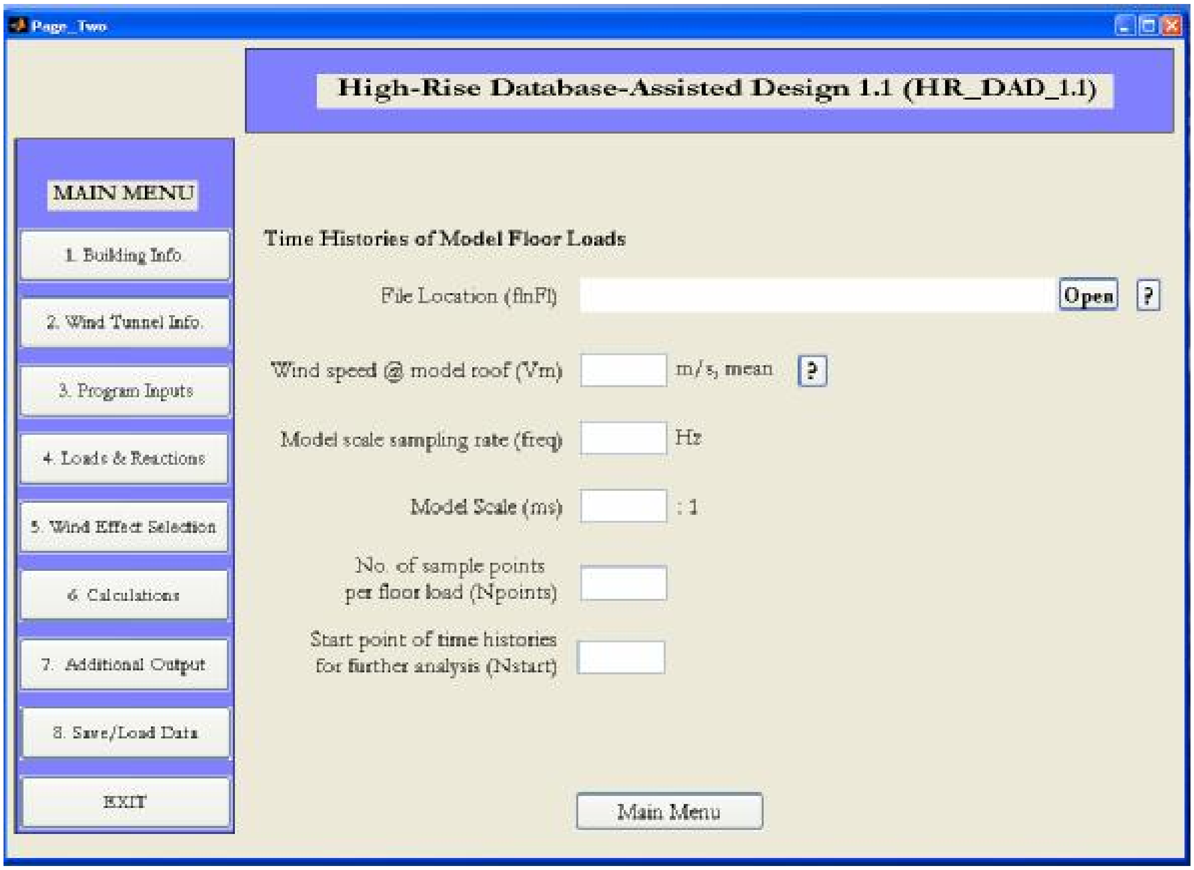
The variable ***evectors*** would therefore take on the following form:

$$\boldsymbol{evectors}=\begin{bmatrix}0.62&0&0\\1&0&0\\0&0.62&0\\0&1&0\\0&0&0.62\\0&0&1\end{bmatrix}$$

and could be saved for example as ModeShapes.mat

END PAGE_ONE INPUT

PAGE_TWO



INPUT:

flnFl = load the MATLAB mat file containing the time histories of the floor loads.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the time histories must be named **F**. The mat file containing the variable **F** can then be saved under any name but must end with "**_XXX**" where **XXX** gives the wind direction, in degrees, from which the wind was blowing in the wind tunnel when the loads where ascertained. An example is shown in Fig. 2 for directions **XXX** = 000 and **XXX** = 035.

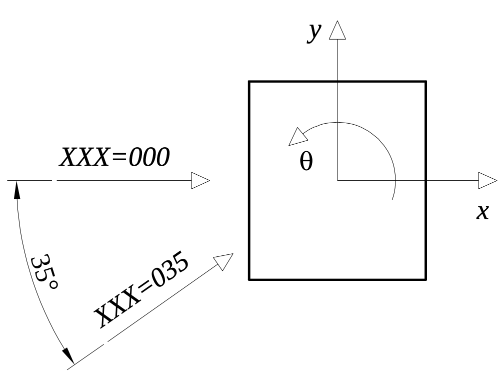


Figure 2.File naming convention used for the floor loads.

For each wind direction a separate mat file is needed containing the relative variable **F**. The variable **F** is a matrix, each line of which contains the time history of a floor load acting in one of the directions *x*, *y* or *q*. The first *N* (*N* is the number of floors) rows correspond to the floor loads acting in the *x*-direction starting from the first floor. The Next *N* rows correspond to the floor loads acting in the *y*-direction while the last *N* rows correspond to the floor loads acting in the *q*-direction. **F** will have a total of *3N* rows.

3D 2-story example: floor loads for running the example can be downloaded from the site www.nist.gov/wind.

Vm = mean wind speed in m/s at roof of the model during the wind tunnel tests.

3D 2-story example: 20m/s

freq = sampling frequency used in the wind tunnel.

3D 2-story example: 250Hz

ms = scale of the model used in the wind tunnel tests.

3D 2-story example: if the scale was 1/500 then the correct number to input is 500.

Npoints = total number of points that make up the time histories of the floor loads.

3D 2-story example: if 30s of data is recoded during the wind tunnel tests with a sampling frequency (**freq**) of 250Hz, then the time histories of the floor loads will have a total of 7504 points.

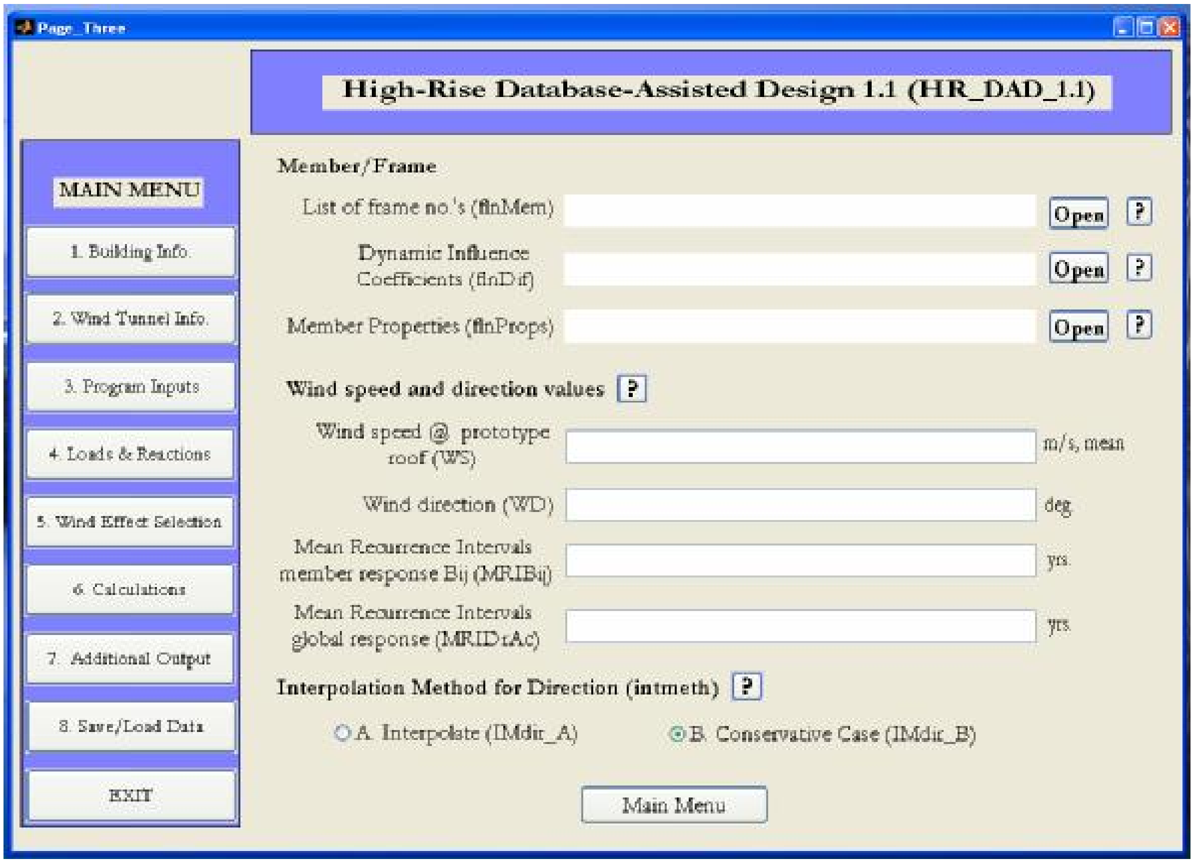
Nstart = number of points to be cut from beginning of the time histories during the analysis.

3D 2-story example: 200 points is reasonable.

Note: Numerical integration needs a certain number of points before it stabilizes. Therefore a certain number of initial points should be cut form the solution of the equations of motion before estimating the response parameters.

END PAGE_TWO INPUT

PAGE_THREE



INPUT:

flnMem = load the list of members composing the structure.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the list of members must be named **mem_list**. The mat file containing the variable **mem_list** can then be saved under any name. The variable **mem_list** is a matrix with two rows and number of columns equal to the number of members that make up the structure. The first row contains the member numbers while the second row contains a label identifying whether the member is a beam, column or diagonal. The labels must be a string of characters that begin with the letters "B" if the element is a Beam, "C" if the element is a column or "D" if the element is a diagonal.

3D 2-story example: the structure has a total of 8 members. The variable **mem_list** would have the following appearance:

$$\textbf{mem\_list} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ "C" & "C" & "C" & "C" & "C" & "C" & "C" & "C" \end{bmatrix}$$

and could be saved for example as members_list.mat

flnDif = load the mat file containing the influence coefficients.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the influence coefficients must be named **dif**. The mat file containing the variable **dif** can then be saved under any name. The variable **dif** is a 3D array. The following description of **dif** will be in reference to the member shown in figure 3.

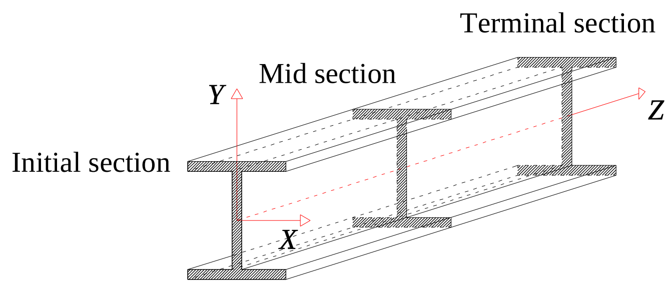


Figure 3. ith member of the structure.

Each face of **dif** contains the influence coefficients associated with the six internal forces (axial force, shear forces and bending moments in the local *X* and *Y* directions and torsion) that can arise in the initial, mid or terminal sections of a given member due to a unit force applied to the center of mass of a given floor in one of the directions *x*, *y* or *q*. In particular, the first column contains the axial forces, second and third columns the *X* and *Y* shears, fourth column the torsion and the fifth and sixth columns the *X* and *Y* bending moments. The first *N* rows (*N* is the number of floors of the building) are the internal forces in the initial section due to unit forces applied to the centers of mass of each floor in the x-directions starting from the 1st floor. The next *N* rows are due to unit forces in the y-direction while successive *N* rows are due to unit torques applied to the centers of mass. This makes up a total of 3*N* rows for the influence coefficients of the initial section. The following 6*N* rows are the influence coefficients for the mid and terminal sections making a total of 9N rows. The index of each face identifies the member. Indeed the *i*th face *must* correspond to the member described in the *i*th column of the variable **mem_list**.

3D 2-story example: the structure in figure 1 has 8 members all of which are columns. Assume them to be steel tubular members with an outside dimension of 25cm and flange thickness of 6mm. Also assume the steel to have an elastic modulus of 200,000 N/mm^2. The next paragraph will explain how to calculate the first face of the array **dif** for this example building.

The first face will coincide with the element in the first column of the variable **mem_list**. It will therefore coincide with member 1. Because the structure has 2 floors, the first face will have 18 rows. The easiest way to calculate the influence coefficients is by constructing a model of the building in a commercial software analysis program, e.g., SAP2000. Within this environment it is then possible to define 3*N* = 6 (*N* is the number of floors which in this case is 2) load cases, one for each unit force applied to the centers of mass of each floor. By running the analysis for the 6 load cases and exporting the results in Microsoft Excel format, tables similar to the one shown in Table 1 are obtained, where the load cases have been named influx1, influx2,…,influz2. The influence coefficients are simply the internal forces highlighted in red. To create the MATLAB mat file simply read the relative rows and columns into MATLAB and construct the array.

Table 1. SAP2000 Excel spreadsheet output for member 1.

TABLE: Element Forces - Frames									
Frame	Station	Output Case	Case Type	P	V2	V3	T	M2	M3
Text	m	Text	Text	N	N	N	N-m	N-m	N-m
1	0	influx1	LinStatic	0	0.25	0	0	0	0.5
1	0	influx2	LinStatic	0	0.25	0	0	0	0.5
1	0	influy1	LinStatic	0	0	0.25	0	0.5	0
1	0	influy2	LinStatic	0	0	0.25	0	0.5	0
1	0	influz1	LinStatic	0	0.039	-0.039	0.011	-0.079	0.079
1	0	influz2	LinStatic	0	0.039	-0.039	0.011	-0.079	0.079
1	2	influx1	LinStatic	0	0.25	0	0	0	0
1	2	influx2	LinStatic	0	0.25	0	0	0	0
1	2	influy1	LinStatic	0	0	0.25	0	0	0
1	2	influy2	LinStatic	0	0	0.25	0	0	0
1	2	influz1	LinStatic	0	0.039	-0.039	0.011	0	0
1	2	influz2	LinStatic	0	0.039	-0.039	0.011	0	0
1	4	influx1	LinStatic	0	0.25	0	0	0	-0.5
1	4	influx2	LinStatic	0	0.25	0	0	0	-0.5
1	4	influy1	LinStatic	0	0	0.25	0	-0.5	0
1	4	influy2	LinStatic	0	0	0.25	0	-0.5	0
1	4	influz1	LinStatic	0	0.039	-0.039	0.011	0.079	-0.079
1	4	influz2	LinStatic	0	0.039	-0.039	0.011	0.079	-0.079

The mat file containing the variable **dif** can be saved, for example, as dif_all.mat

flnProps = load the mat file containing the section properties.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the list of members must be named **props**. The mat file containing the variable **props** can then be saved under any name. The variable **props** is a matrix with first column coinciding with the first row of **mem_list** and therefore contains the member numbering scheme. Each row contains the strength capacities of the member identified in the first column of the row. In particular the second column of each row contains the maximum tensile strength, the third column the maximum compressive strength while the forth and fifth columns the bending strengths in the two local X and Y directions.

3D 2-story example: The variable **props** has the following appearance:

$$\mathbf{props} = \begin{bmatrix} 1 & 1453525 & 1308172 & 133024 & 133024 \\ 2 & 1453525 & 1308172 & 133024 & 133024 \\ 3 & 1453525 & 1308172 & 133024 & 133024 \\ 4 & 1453525 & 1308172 & 133024 & 133024 \\ 5 & 1453525 & 1308172 & 133024 & 133024 \\ 6 & 1453525 & 1308172 & 133024 & 133024 \\ 7 & 1453525 & 1308172 & 133024 & 133024 \\ 8 & 1453525 & 1308172 & 133024 & 133024 \end{bmatrix}$$

The mat file containing the variable **props** could be saved as frame_properties.mat

WS = wind speeds at roof height of the full scale building for which the response is desired.

3D 2-story example: for example, if 5m/s to 50m/s with 5m/s increments are desired then the following vector would be entered:

[5 10 15… 45 50]

Note: The variable must be input using square brackets as shown. Please note that the following notation maybe used [5:5:50].

WD = wind directions for which the response is desired.

3D 2-story example: for example, if 0°, 10°, 20°… 350° and 360° are desired, then the following vector would be entered:

[0 10 20…350 360]

Note 1: The angles input in WD must be the same as or a subset of the angles for which the wind loading is known.

Note 2: The variable must be input using square brackets as shown. Please note that the following notation may be used [0:10:360].

MRIBij = The Mean Recurrence Intervals for which the peak demand/capacity ratios are wanted.

3D 2-story example: for example, if the peak demand/capacity ratios for Mean Recurrence Intervals of 20 and 50 years are desired, then the following vector would be entered:

[20 50]

Note: The variable must be input using square brackets as shown.

MRIDrAc = The Mean Recurrence Intervals for which the peak inter-story drift and top floor acceleration is desired.

3D 2-story example: for example, if the peak inter-story drift and top floor acceleration for Mean Recurrence Intervals of 10 and 20 years are desired, then the following vector would be entered:

[10 20]

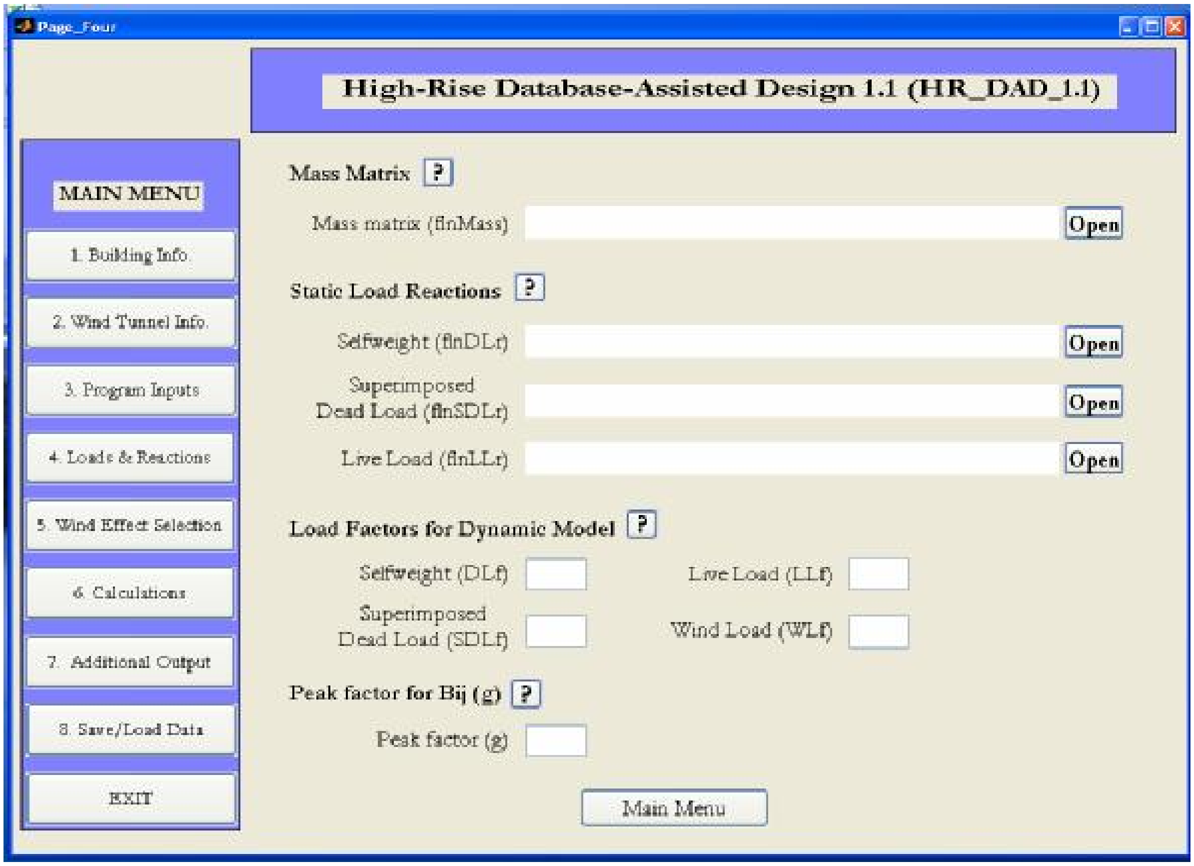
Note: The variable must be input using square brackets as shown.

intmeth = choose the interpolation method that will be implemented during the calculation of the responses with given Mean Recurrence Intervals.

Note: The methods will give similar results. However method A will give more accurate results but take longer than method B, which tends to be more conservative.

END PAGE_THREE INPUT

PAGE_FOUR



INPUT:

flnMass = load the mass and moments of inertia associated with each floor.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the mass and moments of inertia must be named **mass**. The mat file containing the variable **mass** can then be saved under any name. The variable **mass** is a column vector. The first three rows are associated with the first floor. In particular the first entry is the mass in the *x*-direction, the second entry is the mass in the *y*-direction, while the third entry is the moment of inertia with respect to the center of mass. The next 3 rows are associated with the second floor and so forth.

3D 2-story example: considering for both floors a mass in the *x* and *y* directions of 30000kg and moment of inertia of 310000kg m^2, the variable **mass** is:

$$\mathbf{mass} = \begin{bmatrix} 30000 \\ 30000 \\ 310000 \\ 30000 \\ 30000 \\ 310000 \end{bmatrix}$$

The mat file containing the variable **mass** could be saved as mass_asc.mat

flnDLr = load file containing the contribution of the dead weight.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the contribution of the dead weight must be named **frames_DL**. The mat file containing the variable **frames_DL** can then be saved under any name. The variable **frames_DL** is a matrix with first column coinciding with the first row of **mem_list** and therefore contains the member numbering scheme. Each row contains the internal forces due to dead weight occurring in the initial, mid and terminal sections of the member identified in the first column of the row. In particular columns 2 to 4 contain the axial force and bending moments in the two local *X* and *Y* directions (Fig. 3) for the initial section. Columns 5 to 7 contain the same information for the mid section, while columns 8 to 10 are dedicated to axial and bending forces occurring in the terminal section.

3D 2-story example: from a SAP2000 or equivalent model of the building it is possible to access the internal forces that occur in the sections of interest of the 8 members composing the structure. This will give the following **frames_DL**:

$$\textbf{frames\_DL} = \begin{bmatrix} 1 & -153606 & 0 & 0 & -152704 & 0 & 0 & -151803 & 0 & 0 \\ 2 & -76803 & 0 & 0 & -75901 & 0 & 0 & -75000 & 0 & 0 \\ 3 & -153606 & 0 & 0 & -152704 & 0 & 0 & -151803 & 0 & 0 \\ 4 & -76803 & 0 & 0 & -75901 & 0 & 0 & -75000 & 0 & 0 \\ 5 & -153606 & 0 & 0 & -152704 & 0 & 0 & -151803 & 0 & 0 \\ 6 & -76803 & 0 & 0 & -75901 & 0 & 0 & -75000 & 0 & 0 \\ 7 & -153606 & 0 & 0 & -152704 & 0 & 0 & -151803 & 0 & 0 \\ 8 & -76803 & 0 & 0 & -75901 & 0 & 0 & -75000 & 0 & 0 \end{bmatrix}$$

The mat file containing the variable **frames_DL** could be saved as frames_DeadLoad.mat

flnLLr = load file containing the contribution of the live loads.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the contribution of the live loads must be named **frames_LL**. The mat file containing the variable **frames_SDL** can then be saved under any name. The variable **frames_SDL** is a matrix with first column coinciding with the first row of **mem_list** and therefore contains the member numbering scheme. Each row contains the internal forces due to superimposed dead load occurring in the initial, mid and terminal sections of the member identified in the first column of the row. In particular, columns 2 to 4 contain the axial force and bending moments in the two local *X* and *Y* directions (Fig. 3) for the initial section. Columns 5 to 7 contain the same information for the mid section, while columns 8 to 10 are dedicated to axial and bending forces occurring in the terminal section.

3D 2-story example: considering as superimposed dead load a distributed load of 0.5KN/m^2 from the SAP2000 or equivalent model of the building it is possible to access the internal forces that occur in the sections of interest of the 8 members composing the structure. This will give the following **frames_SDL**:

$$\textbf{frames\_SDL} = \begin{bmatrix} 1 & -9000 & 0 & 0 & -9000 & 0 & 0 & -9000 & 0 & 0 \\ 2 & -4500 & 0 & 0 & -4500 & 0 & 0 & -4500 & 0 & 0 \\ 3 & -9000 & 0 & 0 & -9000 & 0 & 0 & -9000 & 0 & 0 \\ 4 & -4500 & 0 & 0 & -4500 & 0 & 0 & -4500 & 0 & 0 \\ 5 & -9000 & 0 & 0 & -9000 & 0 & 0 & -9000 & 0 & 0 \\ 6 & -4500 & 0 & 0 & -4500 & 0 & 0 & -4500 & 0 & 0 \\ 7 & -9000 & 0 & 0 & -9000 & 0 & 0 & -9000 & 0 & 0 \\ 8 & -4500 & 0 & 0 & -4500 & 0 & 0 & -4500 & 0 & 0 \end{bmatrix}$$

The mat file containing the variable **frames_SDL** could be saved as frames_SDeadLoad.mat

flnSDLr = load file containing the contribution of the superimposed dead load.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the contribution of the superimposed dead load must be named **frames_LL**. The mat file containing the variable **frames_LL** can then be saved under any name. The variable **frames_LL** is a matrix with first column coinciding with the first row of **mem_list** and therefore contains the member numbering scheme. Each row contains the internal forces due to live loads occurring in the initial, mid and terminal sections of the member identified in the first column of the row. In particular, columns 2 to 4 contain the axial force and bending moments in the two local *X* and *Y* directions (Fig. 3) for the initial section. Columns 5 to 7 contain the same information for the mid section while columns 8 to 10 are dedicated to axial and bending forces occurring in the terminal section.

3D 2-story example: considering as live load a distributed load of 2KN/m^2 from the SAP2000 or equivalent model of the building it is possible to access the internal forces that occur in the sections of interest of the 8 members composing the structure. This will give the following **frames_LL**:

$$\textbf{frames\_LL} = \begin{bmatrix} 1 & -36000 & 0 & 0 & -36000 & 0 & 0 & -36000 & 0 & 0 \\ 2 & -18000 & 0 & 0 & -18000 & 0 & 0 & -18000 & 0 & 0 \\ 3 & -36000 & 0 & 0 & -36000 & 0 & 0 & -36000 & 0 & 0 \\ 4 & -18000 & 0 & 0 & -18000 & 0 & 0 & -18000 & 0 & 0 \\ 5 & -36000 & 0 & 0 & -36000 & 0 & 0 & -36000 & 0 & 0 \\ 6 & -18000 & 0 & 0 & -18000 & 0 & 0 & -18000 & 0 & 0 \\ 7 & -36000 & 0 & 0 & -36000 & 0 & 0 & -36000 & 0 & 0 \\ 8 & -18000 & 0 & 0 & -18000 & 0 & 0 & -18000 & 0 & 0 \end{bmatrix}$$

The mat file containing the variable **frames_LL** could be saved as frames_LiveLoad.mat

DLf = dead load factor.

Note: the dead loads, superimposed dead loads, live loads and wind loads must be combined. **HR_DAD_1.1** allows the user to combine these loads by defining load factors which are then used to define appropriate load combinations.

3D 2 story example: using the ASCE 7-05 Standard load combination:

1.2D + 1.0L + 1.6W

in which D is the dead and superimposed dead loads, L is the live load while W is the wind load, the appropriate value is therefore 1.2.

SDLf = superimposed dead load factor.

3D-2 story example: see above under **DLf** .

LLf = live load factor.

3D-2 story example: see above under **DLf**.

WLf = wind load factor.

3D-2 story example: see above under **DLf**.

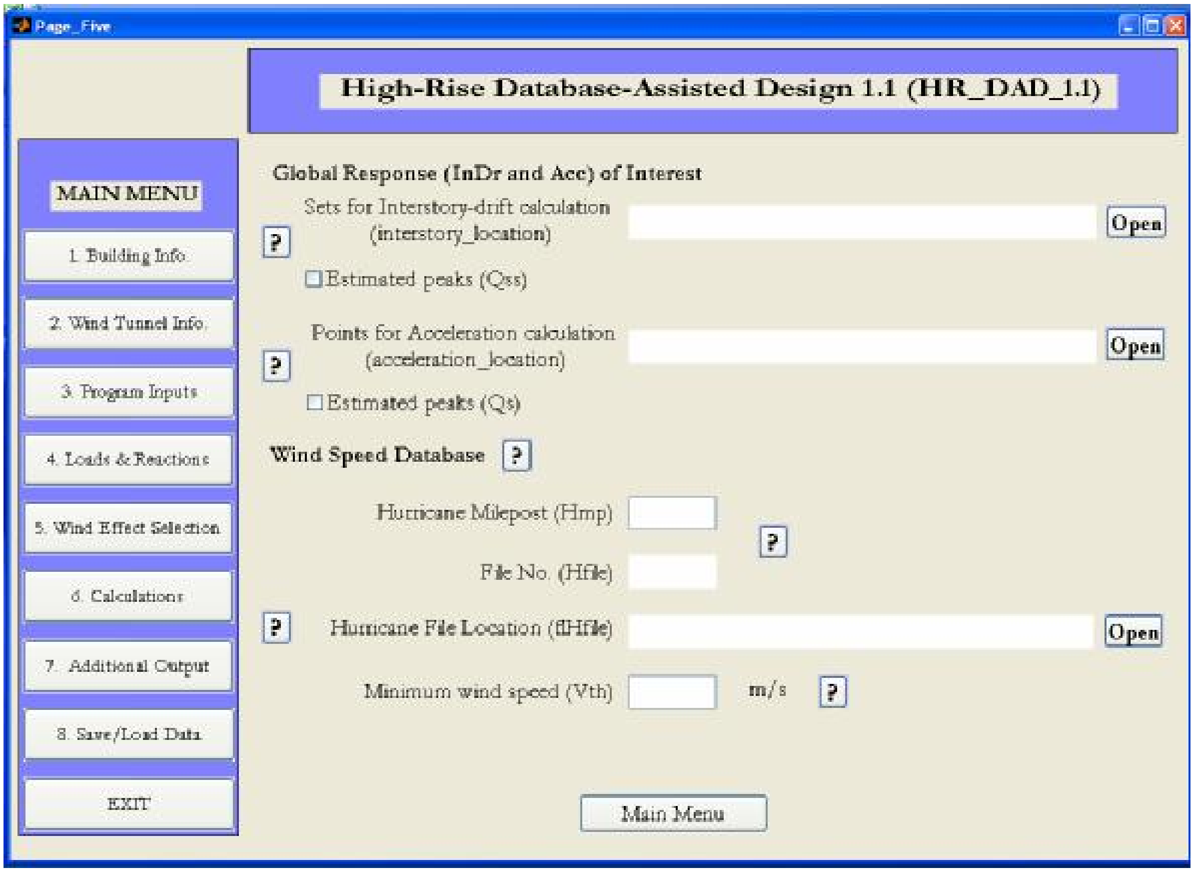
g = local response peak factor.

Note: the local response peak factor is used to calculate the peak internal forces occurring in the various members. Appropriate values of this parameter are between 3 and 4.

3D 2 story example: an appropriate value would be 3.5.

END PAGE_ FOUR INPUT

PAGE_FIVE



INPUT:

interstory_location = load the file containing the position of the column line where the inter-story drift is desired.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the position of the column line must be named **interstory_location**. The mat file containing the variable **interstory_location** can then be saved under any name. The variable **interstory_location** is a matrix, the first 2 rows of which contain the x and y coordinates of the column line with respect to the center of mass of each floor. The third and last column contains the height of the story. The first row contains information relating to the first floor, while the second row stores information related to the second floor and so forth. Successive column lines are appended as an extra N rows (where N is the number of floors).

3D 2-story example: wanting to consider the column line formed by members 7-8 and 5-6 (Fig. 1) the variable **interstory_location** is:

$$\textbf{interstory\_location} = \begin{bmatrix} 3 & 3 & 4 \\ 3 & 3 & 4 \\ 3 & -3 & 4 \\ 3 & -3 & 4 \end{bmatrix}$$

The mat file containing the variable **interstory_location** could be saved as Interstory_Drift_Input.mat

Qss = choose whether to consider observed peaks or peaks estimated as indicated in Sect. III B, www.nist.gov/wind.

Note: the default is to consider the observed peaks of the inter-story drift time histories.

acceleration_location = load the file containing the positions of the points belonging to the top floor where the peak acceleration is desired.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the position of the points must be named **acceleration_location**. The mat file containing the variable **acceleration_location** can then be saved under any name. The variable **acceleration_location** is a matrix, the first 2 rows of which contain the x and y coordinates of the point with respect to the center of mass of the top floor. The first row contains information relating to the first point while the second row stores information related to the second point and so forth.

3D 2-story example: If one of the corner points is considered, for instance the point with the coordinates $x = 3$ and $y = 3$, the variable **acceleration_location** is:

$$\textbf{acceleration\_location} = \begin{bmatrix} 3 & 3 \end{bmatrix}$$

The mat file containing the variable **acceleration_location** could be saved as Acceleration_Input.mat

Qs = choose whether to consider observed peaks or peaks estimated as indicated in Sect. III B, www.nist.gov/wind.

Note: the default is to consider the observed peaks of the top floor acceleration time histories.

The following pertains to the simulated 999 extreme wind events provided for a large number of locations (mileposts) along the Gulf of Mexico and North Atlantic coast. This simulated data is publicly available at http://www.nist.gov/wind by following the links for extreme wind data sets.

Hmp = hurricane milepost

3D 2-story example: considering the building located in Miami the milepost is 1450.

flHfile = folder location of the database of simulated hurricanes.

3D 2-story example: after downloading the simulated hurricane database place it into any folder on your machine and point **flHfile** to this folder.

Vth = minimum wind speed under which the response is no longer desired

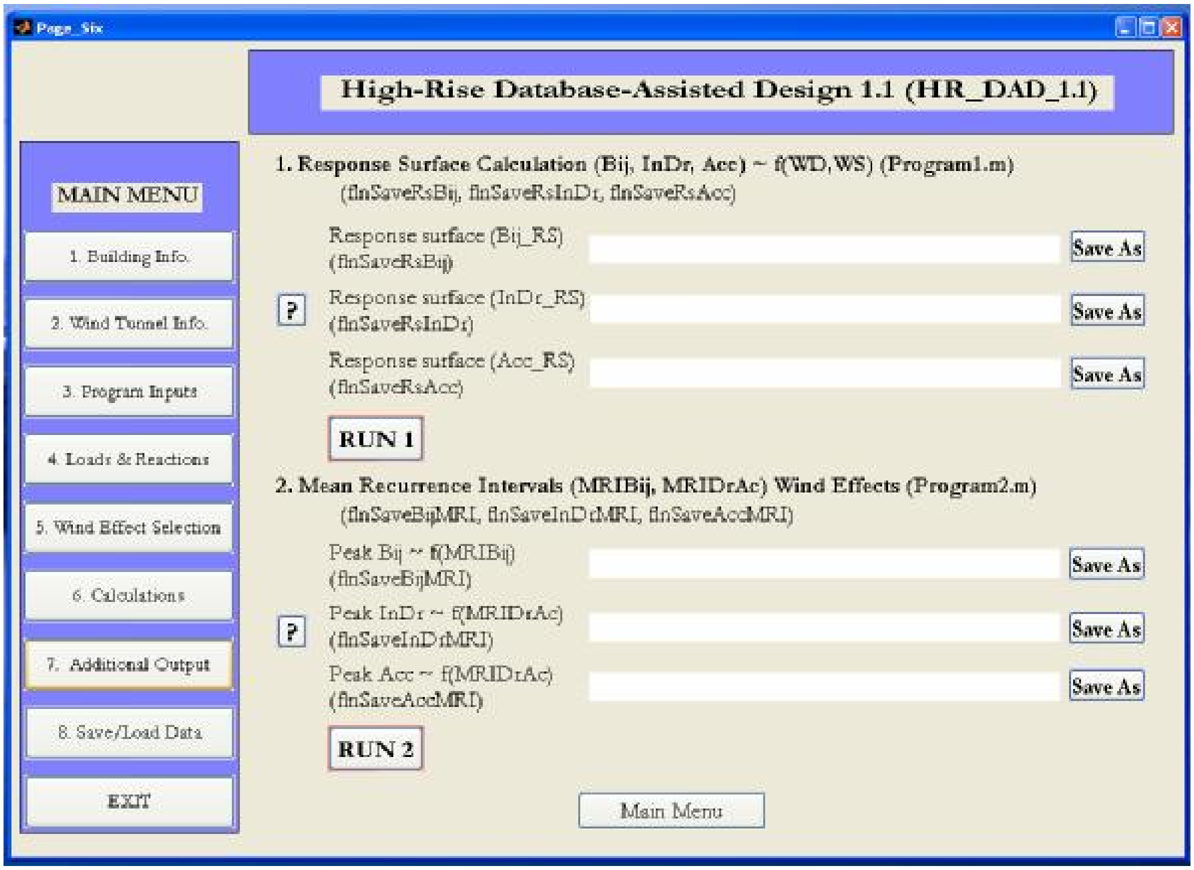
Note: If a low minimum wind speed is considered this will lengthen the calculation of the responses with a specified MRI. However the higher the minimum wind speed is the greater the possibility of not considering a critical wind effect is.

3D 2-story example: a reasonable minimum speed is 15 m/s

Note: In general several types of wind need to be considered – see Sect. 6, item 4 in this report. Software for inclusion of up to three types of wind is being developed. See also Grigoriu (2008).

END PAGE_ FIVE INPUT

PAGE_SIX



OUTPUT:

flnSaveRsBij = specify file location and name where the member response surfaces will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the response surfaces will be named **Bij_RS**. The mat file containing the variable **Bij_RS** can then be saved under any name. The variable **Bij_RS** is a 3D array, each face of which represents the response surface of a specific member. The file structure is schematically represented in Fig. 4.

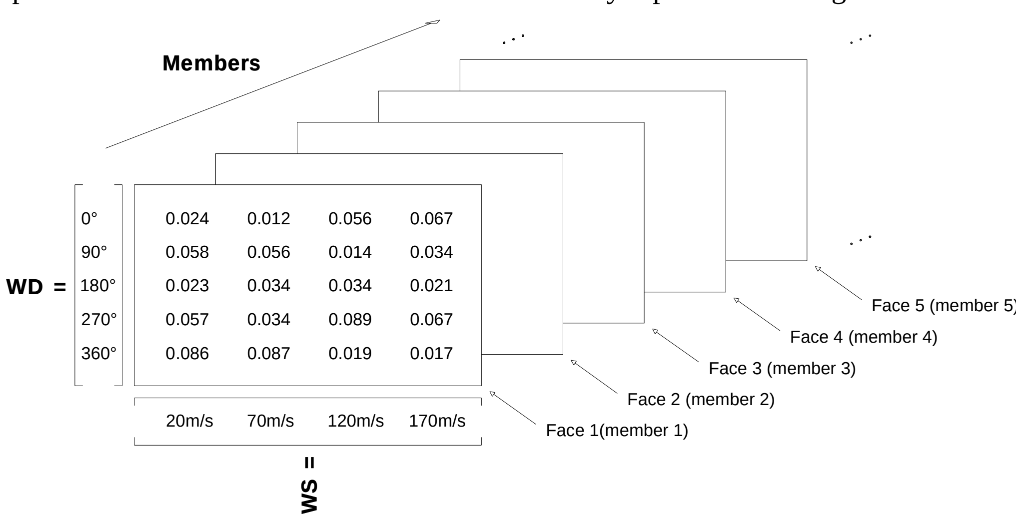


Figure 4. **Bij_RS** *file structure.*

In particular the index of each face corresponds to the column index of **mem_list** and therefore identifies the member. An element of a specific face identified by the indexes *(i,j)* corresponds to the peak demand/capacity ratio for the wind direction given by the *i*th element of **WD** and the wind speed given by the *j*th element of **WS**. Therefore each face will have a number of rows coinciding with the number of wind directions in **WD**, and a number of columns coinciding with the number of wind speeds in **WS**.

3D 2-story example: plotting the first face of **Bij_RS** gives a graphical representation of the response surface associated with member 1. The plot is shown in figure 5.

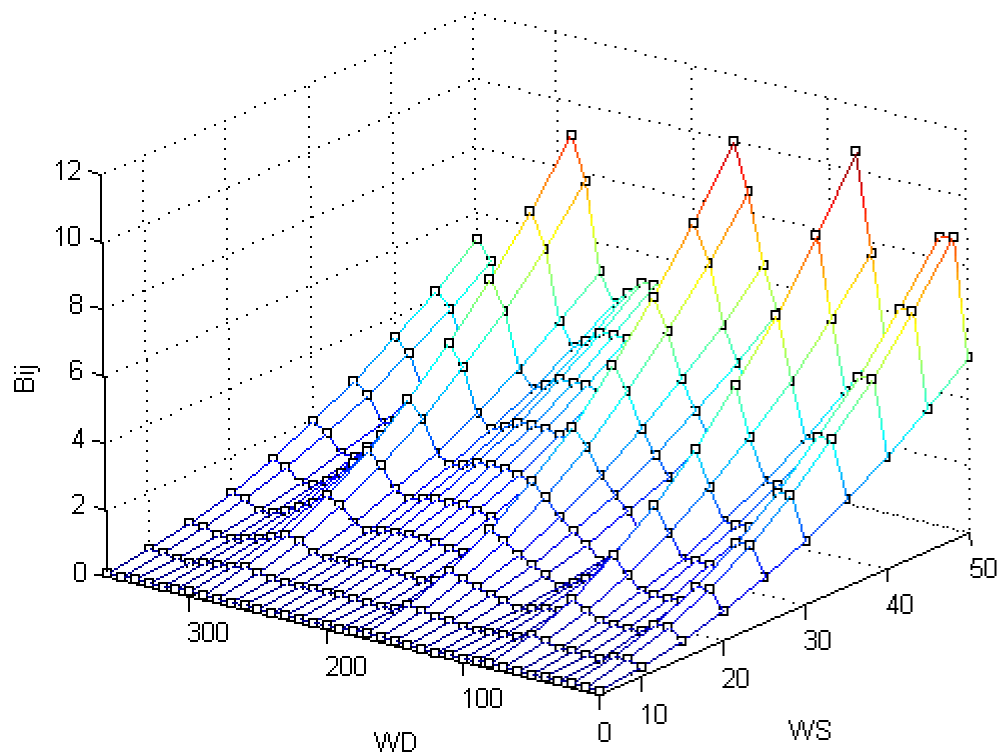


Figure 5. response surface associated with member 1.

flnSaveRsInDr = specify file location and name where the inter-story response surfaces will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the response surfaces will be named **InDr_RS_set_X** where X depends on the column number under consideration. The mat file containing the variable **InDr_RS_set_X** can then be saved under any name. The variable **InDr_RS_set_X** is a 3D array, each face of which represents the response surface in a specific direction (*x* or *y*). In particular the file is arranged so that the first *N* faces (*N* is the number of floors of the building) of the array are associated with the response in the *x*-direction while the next N faces are associated with the *y*-direction. The structure of the file is schematically shown in figure 6. An element of a specific face identified by the indexes *(i,j)* corresponds to the peak inter-story drift for the wind direction given by the *i*th element of **WD** and the wind speed given by the *j*th element of **WS**. Therefore each face will have a number of rows coinciding with the number of wind directions in **WD**, and a number of columns coinciding with the number of wind speeds in **WS**.

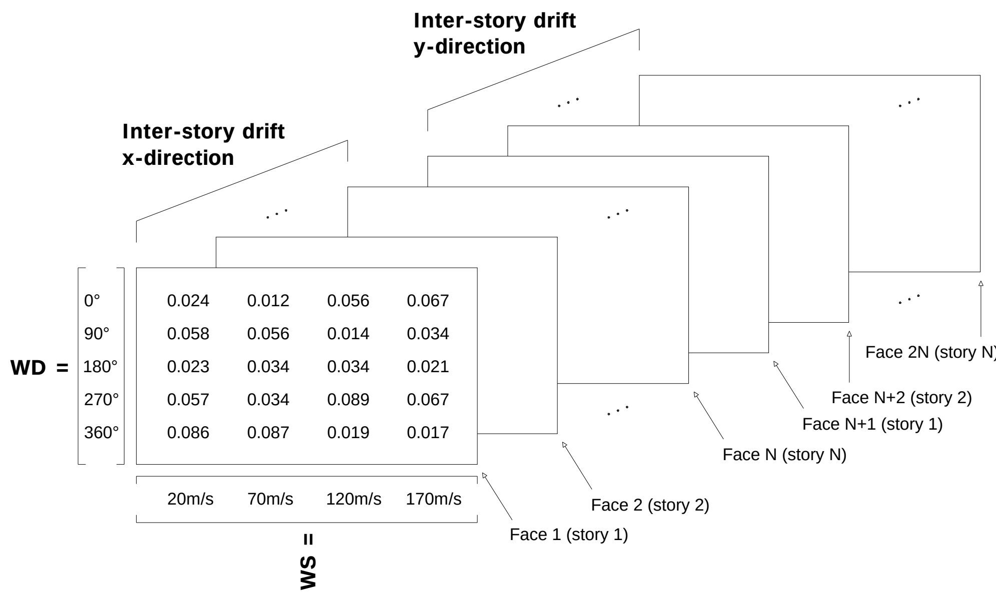


Figure 6. **InDr_RS_set_X** *file structure.*

3D 2-story example: plotting the second face of **InDr_RS_set_1** gives a graphical representation of the response surface associated with the inter-story drift in direction *x* of column line 1. The plot is shown in Fig. 7.

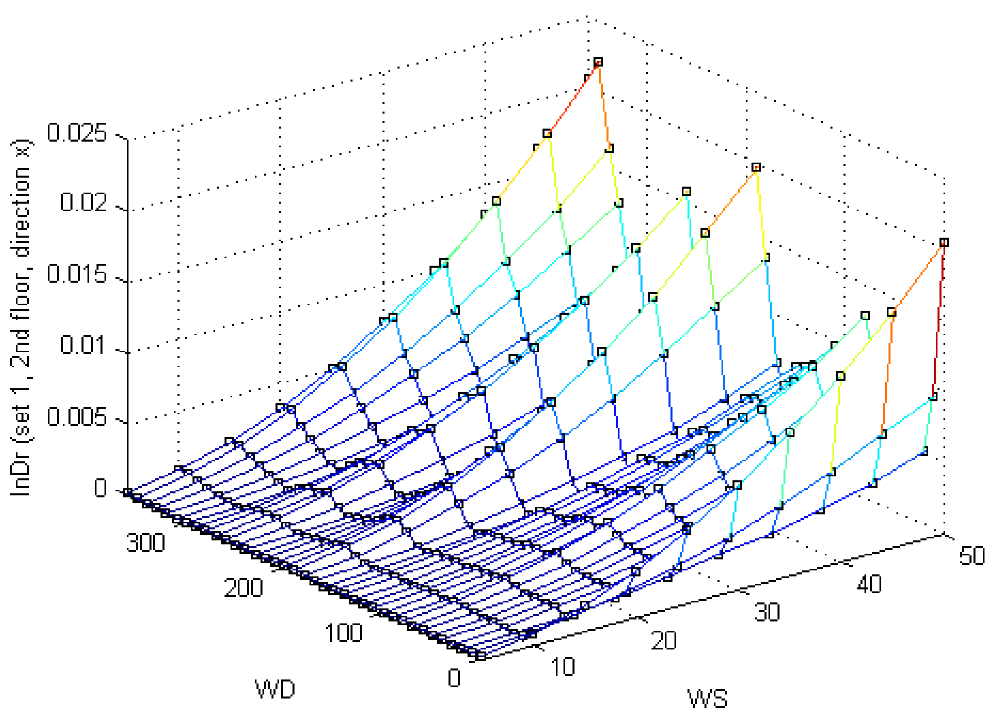


Figure 7. 2nd x-direction floor response surface of set 1

flnSaveRsAcc = specify file location and name where the top floor acceleration response surfaces will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the response surfaces will be named **Acc_RS_point_X** where X depends on the point belonging to the top floor. The mat file containing the variable **Acc_RS_point_X** can then be saved under any name. The variable **Acc_RS_point_X** is a 3D array, each face of which represents the response surface in a specific direction (*x* or *y*). In particular, the file is arranged so that the first face of the array is associated with the response in the *x*-direction while the next face is associated with the *y*-direction. The structure of the file is schematically shown in Fig. 8. As can be seen an element of a specific face identified by the indexes (*i*,*j)* corresponds to the peak inter-story drift for the wind direction given by the *i*th element of **WD** and the wind speed given by the *j*th element of **WS**. Therefore each face will have a number of rows coinciding with the number of wind directions in **WD**, and a number of columns coinciding with the number of wind speeds in **WS**.

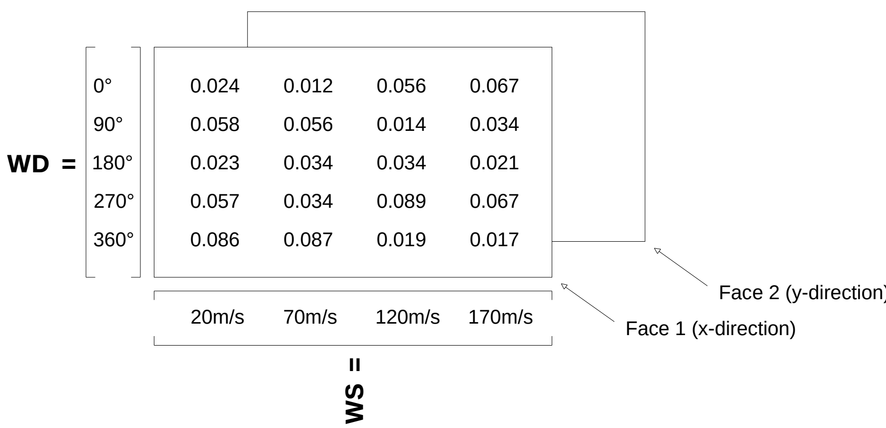


Figure 8. **Acc_RS_point_X** *file structure.*

3D 2-story example: Plotting the faces of **Acc_RS_point_1** gives results similar to those shown for the member response surfaces and inter-story drift.

flnSaveBijMRI = specify file location and name where the member responses with specified Mean Recurrence Intervals (MRIs) will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the member responses with specified MRIs will be named **Bij_MRIs**. The mat file containing the variable **Bij_MRIs** can then be saved under any name. The variable **Bij_MRIs** is a matrix the rows of which contain the peak capacity/demand ratios for a specific MRI. In particular the first row will contain the ratios corresponding to the first MRI contained in **MRIBij**, the second row those corresponding to the second MRI of **MRIBij** and so forth. The index of each column corresponds to the column index of **mem_list** and therefore identifies the member.

3D 2-story example: considering the MRIs input on page 3, **MRIBij =** [20 50], **Bij_MRIs** takes on the following form:

$$\textbf{Bij\_MRIs} = \begin{bmatrix} 1.11 & 0.41 & 1.35 & 0.49 & 1.33 & 0.49 & 1.18 & 0.43 \\ 1.33 & 0.49 & 1.66 & 0.60 & 1.62 & 0.59 & 1.39 & 0.51 \end{bmatrix}$$

Therefore **Bij_MRIs(1,1) =** 1.11 means the for member 1 the maximum demand/capacity ratio with a MRI of 20 years is 1.11.

flnSaveInDrMRI = specify file location and name where the inter-story drift responses with specified Mean Recurrence Intervals (MRIs) will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the inter-story responses with specified MRIs will be named **InDr_MRIs_set_X** where X depends on the column number under consideration. The mat file containing the variable **InDr_MRIs_set_X** can then be saved under any name. The variable **InDr_MRIs_set_X** is a 3D array, each face of which represents the inter-story drift response, for column line X, with specified MRI. In particular the first face will contain the responses corresponding to the first MRI contained in **MRIDrAc**, the second face those corresponding to the second MRI of **MRIDrAc** and so forth. Each face has two columns corresponding to the *x* and *y* directions, respectively, and *N* rows which correspond to the *N* floors.

3D 2-story example: Considering the MRIs input on page 3, **MRIDrAc =** [10 20], the first face of **InDr_MRIs_set_1** takes on the following form:

$$\textbf{InDr\_MRIs\_set\_1}\ (\,face\ 1) = \begin{bmatrix} 0.025 & 0.026 \\ 0.015 & 0.016 \end{bmatrix}$$

Therefore **InDr_MRIs_set_1(1,1, face 1) =** 0.025 means that the peak inter-story drift in direction x of column line 1, floor 1 with MRI of 10 years is 0.025. In direction y floor 1, it is **InDr_MRIs_set_1(1,2, face 1) =** 0.026 and so forth.

flnSaveAccMRI = specify file location and name where the top floor acceleration responses with specified Mean Recurrence Intervals (MRIs) will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the top floor acceleration responses with specified MRIs will be named **Acc_MRIs_point_X** where X depends on the point belonging to the top floor. The mat file containing the variable **Acc_MRIs_point_X** can then be saved under any name. The variable **Acc_MRIs_point_X** is a 3D array, each face of which represents the top floor acceleration response, for point X, with specified MRI. In particular the first face will contain the responses corresponding to the first MRI contained in **MRIDrAc**, the second face those corresponding to the second MRI of **MRIDrAc** and so forth. Each face has two columns corresponding to the *x* and *y* directions.

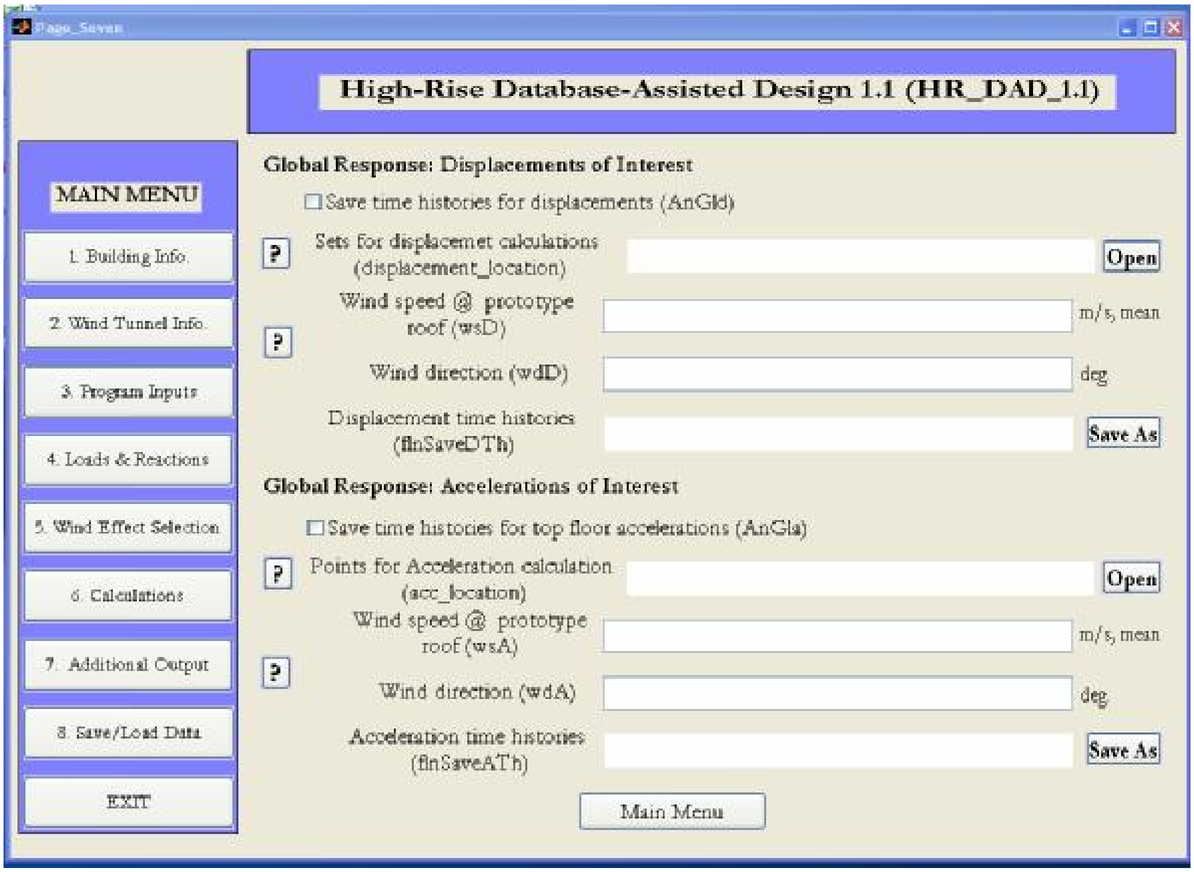
3D 2-story example: Considering the MRIs input on page 3, **MRIDrAc =** [10 20], the first face of **Acc_MRIs_point_1** takes on the following form:

$$\textbf{Acc\_MRIs\_point\_1}\ (\,face\ 1) = \begin{bmatrix} 0.0319 & 0.0326 \end{bmatrix}$$

Therefore **Acc_MRIs_point_1 (1,1, face 1) =** 0.0319 means that the peak top floor acceleration in direction *x* of point 1 with MRI of 10 years is 0.0329, while in direction *y* it is **Acc_MRIs_point_1 (1,2, face 1) =** 0.0326 and so forth.

END PAGE_ SIX OUTPUT

PAGE_SEVEN (Additional output)



DISPLACEMENTS:

AnGld = choose whether to save select time histories of displacements.

displacement_location = load the file containing the position of the column line where the time histories of the displacements are desired.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the position of the column line must be named **displacement_location**. The mat file containing the variable **displacement_location** can then be saved under any name. The variable **displacement_location** is a matrix, the first 2 columns of which contain the x and y coordinates of the column line with respect to the center of mass of each floor. The first row contains information relating to the first floor while the second row stores information related to the second floor and so forth. Successive column lines are appended as an extra N rows (where N is the number of floors).

3D 2-story example: If the column line formed by members 7-8 (Fig. 1) is considered, the variable **displacement_location** is:

$$\textbf{displacement\_location} = \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}$$

The mat file containing the variable **displacement_location** could be saved as Displacement_Input.mat

wsD = wind speeds at roof height of the full scale building for which the displacement time histories are desired.

3D 2-story example: For example, if 30m/s, 40m/s are desired then the following vector would be entered:

[30 40]

Note: The variable must be input using square brackets as shown and must be *equal or a subset* of **WS**

wdD = wind directions for which the displacement time histories are desired.

3D 2-story example: For example, if 40° and 180° are desired, then the following vector would be entered:

[40 180]

Note 1: The variable must be input using square brackets as shown and must be *equal to or a subset* of **WD**

Note 2: The variables **wsD** and **wdD** *must* have the same lengths. Indeed the time histories of the displacements are calculated for pairs of wind speeds and directions.

flnSaveDTh = specify file location and name where the time histories of the displacements will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the displacements time histories will be named **disp_set_X_wdD_wsD** where X depends on the column line while **wdD** and **wsD** will take the values associated with the time history. Therefore for each wind direction and speed a separate mat file will be generated. The mat file containing the variable **disp_set_X_wdD_wsD** can then be saved under any name but will end with **_set_X_wdD_wsD** indentifing the column line and wind speed and direction. The variable **disp_set_X_wdD_wsD** is a 3D array, each face of which represents the displacement time histories in a particular direction. The first face is associated with the *x*-direction, the second with the *y*-direction while the third and last is associated with the rotation around the *z*-axis. The faces are arranged so that each row contains a time history. The first row is associated with the first floor, the second row with the second floor and so forth.

3D 2-story example: Having considered a single column line (members 1-2, Fig. 1) with two sets of wind speeds and directions, two mat files will be saved containing the variables **disp_set_1_40_30** and **disp_set_1_180_40**. If it is desired to display the time histories associated with the displacements in the *x*-direction at the top of column line 1, for a wind speed of 40m/s and direction of 30°, it is simply necessary to plot **disp_set_1_40_30(2,:,1);** this constitutes the second row of the first face. The time history is shown in Fig. 9.

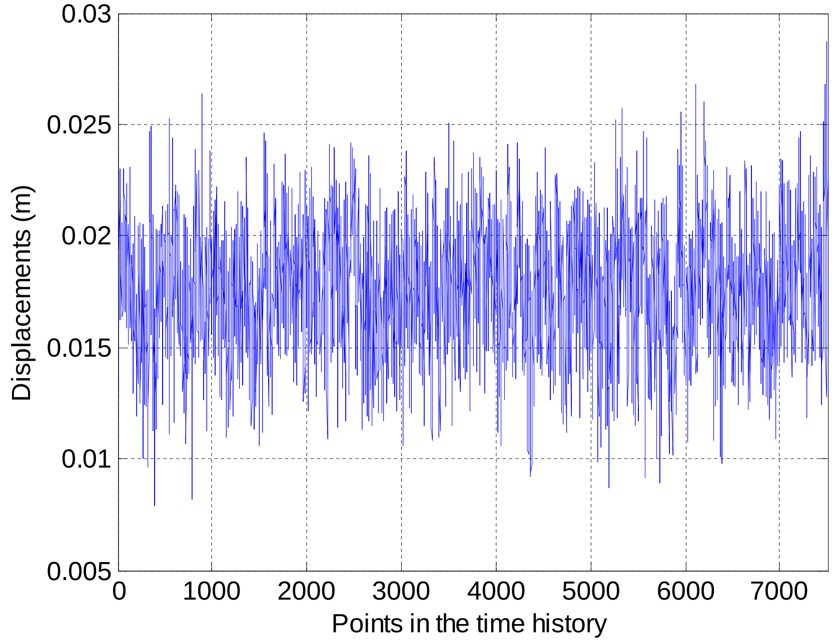


Figure 9. **disp_set_1_40_30(2,:,1)**, *2nd floor x-direction displacement time history for column line 1.*

TOP FLOOR ACCELERATION:

AnGla = choose whether to save select time histories of top floor accelerations.

acc_location = load the file containing the positions of the points where the time histories of the top floor accelerations are desired.

File structure: The file can be constructed in MATLAB. The variable in the mat file containing the position of the points must be named **acc_location**. The mat file containing the variable **acc_location** can then be saved under any name. The variable **acc_location** is a matrix, the first 2 columns of which contain the *x* and *y* coordinates of the points with respect to the center of mass of the top floor. The first row contains information relating to the first point while the second row stores information related to a possible second point and so forth.

3D 2-story example: If the point above member 8 (Fig. 1) is considered, the variable **acc_location** is:

$$\mathbf{acc\_location} = \begin{bmatrix} 3 & 3 \end{bmatrix}$$

The mat file containing the variable **acc_location** could be saved as Acc_Input.mat

wsA = wind speeds at roof height of the full scale building for which the top floor acceleration time histories are desired.

3D 2-story example: For example, if 35m/s, 45m/s are desired then the following vector would be entered:

[35 45]

Note: The variable must be input using square brackets as shown and must be *equal to or a subset* of **WS**

wdA = wind directions for which the top floor acceleration time histories are desired.

3D 2-story example: For example, if 70° and 250° are desired, then the following vector would be entered:

[70 250]

Note 1: The variable must be input using square brackets as shown and must be *equal to or a subset* of **WD**

Note 2: The variables **wsA** and **wdA** *must* have the same lengths. Indeed the time histories of the displacements are calculated for pairs of wind speeds and directions.

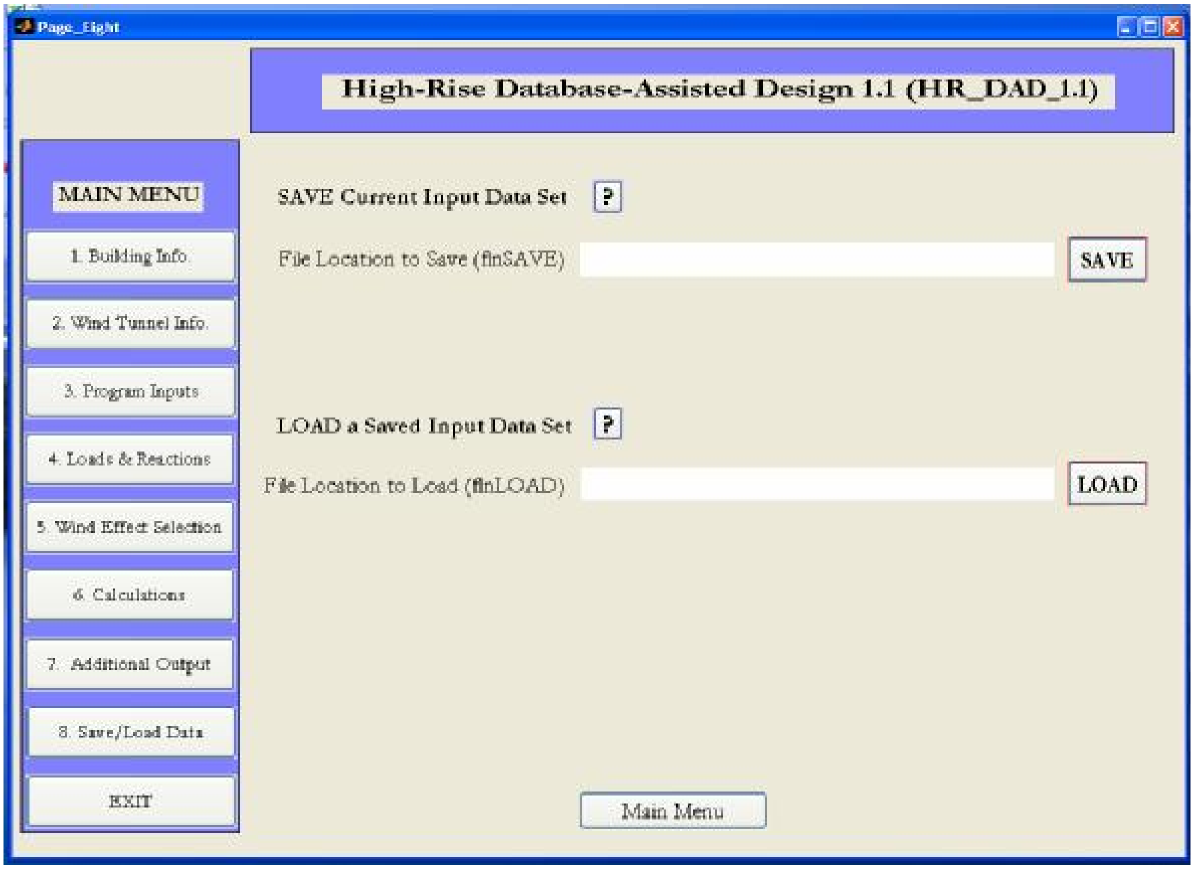
flnSaveATh = specify file location and name where the time histories of the top floor accelerations will be saved.

File structure: The file that will be saved is a MATLAB mat file. The variable in the mat file containing the top floor acceleration time histories will be named **Top_floor_acc_wdD_wsD** where **wdD** and **wsD** will take the values associated with the particular time history. Therefore for each wind direction and speed a separate mat file will be generated. The mat file containing the variable **Top_floor_acc_wdD_wsD** can then be saved under any name but will end with **_wdD_wsD** indentifing the wind speed and direction. The variable **Top_floor_acc_wdD_wsD** is a 3D array, each face of which represents the top floor acceleration time histories in a particular direction. The first face is associated with the *x*-direction, the second with the *y*-direction while the third and last is associated with the rotation around the *z*-axis. The faces are arranged so that each row contains a time history. The first row is associated with the first point input form **acc_location**, the second row with a possible second point, and so forth.

3D 2-story example: wanting to evaluate the time history of the top floor acceleration in direction x of the point input in **acc_location** for a wind direction 70° and speed 35m/s, it is simply necessary to evaluate **Top_floor_acc_70_35(1,:,1)**.

END PAGE_ SEVEN

PAGE _EIGHT (Additional output)



DISPLACEMENTS:

flnSAVE = specify file location and name where the all the input information loaded through pages one to seven will be saved.

File structure: The file is a MATLAB mat file that contains all the necessary information input at the time the file was saved

flnLOAD = load a file containing input information for running **HR_DAD_1.1**.

File structure: The file is a MATLAB mat file generated during the definition of **flnSAVE**.

END PAGE_EIGHT